In the Claims:

1. (Currently Amended) A method for a middle-tier server to impersonate a client to a plurality of servers, the method comprising:

obtaining a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers;

providing the common nonce to the client;

receiving the common nonce signed by the client at the middle-tier server; and

providing the signed common nonce as a signature for transactions from the client to the plurality of servers so as to authenticate the client to the plurality of servers.

2. (Original) The method of Claim 1, wherein the step of obtaining a common nonce comprises the step of generating a common nonce based on information obtained from each of the plurality of servers.

3. (Original) The method of Claim 2, wherein the step of generating a common nonce comprises the steps of:

obtaining pre-nonce contributions from the plurality of servers;

combining the pre-nonce contributions to provide a single pre-nonce token; and

providing the common nonce based on the pre-nonce token.

4. (Original) The method of Claim 3, wherein the step of providing the common nonce comprises reducing the pre-nonce token to provide the common nonce.

5. (Original) The method of Claim 3, wherein the step of combining the pre-nonce contributions to provide a single pre-nonce token comprises concatenating the pre-nonce contributions.

6. (Original) The method of Claim 4, wherein the step of reducing the pre-nonce token to provide the common nonce comprises the step of hashing the pre-nonce token utilizing a one-way hash function so as to provide the common nonce.

7. (Original) The method of Claim 3, wherein the step of obtaining pre-nonce contributions comprises the steps of:

requesting a pre-nonce contribution from each of the plurality of servers; and

receiving the pre-nonce contributions from the plurality of servers.

8. (Original) The method of Claim 7, wherein requesting a pre-nonce contribution comprises sending authenticated requests to the plurality of servers.

9. (Original) The method of Claim 8, further comprising the step of encrypting the authenticated requests sent to the plurality of servers.

10. (Original) The method of Claim 8, wherein the authenticated requests include at least one of an identification of a source of the request, a time stamp and a random number.

11. (Original) The method of Claim 3, wherein the pre-nonce contributions include at least one of an identification of a server of the plurality of servers and a random number.

12. (Original) The method of Claim 3, wherein the pre-nonce contributions are signed with a signature corresponding to a server from which the pre-nonce contribution was obtained, the method further comprising incorporating the signatures in the pre-nonce token.

13. (Original) The method of Claim 3, wherein the pre-nonce contributions are signed with a signature corresponding to a server from which the pre-nonce contribution was obtained, the method further comprising authenticating the signatures of the pre-nonce

contributions and rejecting pre-nonce contributions for which the digital signature is not authentic.

14. (Original) The method of Claim 3, further comprising the steps of:

receiving a transaction identification from a trusted server of the plurality of servers; and

associating the transaction identification with the common nonce.

15. (Original) The method of Claim 14, further comprising the step of tracking use of the common nonce based on the transaction identification.

16. (Original) The method of Claim 3, further comprising the steps of:

associating an expiration time with a pre-nonce contribution; and

determining if the pre-nonce contribution has expired based on its associated expiration time.

17. (Original) The method of Claim 16, further comprising the steps of:

receiving the common nonce at a server of the plurality of servers;

determining a pre-nonce contribution associated with the received common nonce; and

accepting the received common nonce if the associated pre-nonce contribution has not expired.

18. (Original) The method of Claim 3, wherein at least one of the plurality of servers carries out the steps of:

receiving a client certificate;

determining if the client certificate is trusted; and

indicating that the client is not authenticated if the client certificate is not trusted.

19. (Original) The method of Claim 3, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce and a client certificate;

determining if the signature of the signed common nonce corresponds to a signature of the client certificate; and

indicating that the client is not authenticated if the signature of the signed common nonce does not correspond to the signature of the client certificate.

20. (Original) The method of Claim 6, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce, the common nonce and the pre-nonce token;

hashing the received pre-nonce token;

comparing the hashed pre-nonce token to the common nonce;

indicating that the client is not authenticated if the hashed pre-nonce token is different from the common nonce.

21. (Original) The method of Claim 11, wherein at least one of the plurality of servers carries out the steps of:

receiving the pre-nonce token;

determining if the pre-nonce token includes a random number associated with the at least one of the plurality of servers; and

indicating that the client is not authenticated if the pre-nonce token does not include the random number associated with the at least one of the plurality of servers.

22. (Original) The method of Claim 21, wherein at least one of the plurality of servers carries out the steps of:

associating an expiration with the random number associated with the at least one of the plurality of servers; and

indicating that the client is not authenticated if the pre-nonce token does not include a random number associated with the at least one of the plurality of servers which has not expired.

23. (Original) The method of Claim 1, wherein the step of obtaining a common nonce comprises the steps of:

obtaining the common nonce from a party trusted by the middle-tier server and the plurality of servers, the common nonce being signed by the trusted party; and

verifying the signature of the common nonce is the signature of the trusted party.

24. (Original) The method of Claim 23, wherein at least one of the plurality of servers carries out the steps of:

receiving a client certificate;

determining if the client certificate is trusted; and

indicating that the client is not authenticated if the client certificate is not trusted.

25. (Original) The method of Claim 23, wherein at least one of the plurality of servers carries out the steps of:

receiving the signed common nonce and a client certificate;

determining if the signature of the signed common nonce corresponds to a signature of the client certificate; and

indicating that the client is not authenticated if the signature of the signed common nonce does not correspond to the signature of the client certificate.

26. (Currently Amended) A system for a middle-tier server to impersonate a client to a plurality of servers, comprising:

means for obtaining a common nonce associated with each of the plurality of servers <u>from an entity other than the client or the plurality of servers</u>;

means for providing the common nonce to the client;

means for receiving the common nonce signed by the client at the middle-tier server; and

means for providing the signed common nonce as a signature for transactions from the client to the plurality of servers so as to authenticate the client to the plurality of servers.

27. (Currently Amended) A computer program product for a middle-tier server to impersonate a client to a plurality of servers, comprising:

a computer readable media having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code that obtains a common nonce associated with each of the plurality of servers from an entity other than the client or the plurality of servers;

computer readable program code that provides the common nonce to the client;

computer readable program code that receives the common nonce signed by the client at the middle-tier server; and

computer readable program code that provides the signed common nonce as a signature for transactions from the client to the plurality of servers so as to authenticate the client to the plurality of servers.

28. (Currently Amended) A method of authenticating a client, comprising:

receiving at a server of a plurality of servers, a common nonce which is associated with each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being signed by the client; and

authenticating the client based on the received signed common nonce.

29. (Original) The method of Claim 28, wherein the common nonce is provided by a trusted third party.

30. (Original) The method of Claim 28, wherein the common nonce is generated based on information provided by each of the plurality of servers.

31. (Currently Amended) A system for authenticating a client, comprising:

means for receiving at a server of a plurality of servers, a common nonce which is associated with each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being signed by the client; and

means for authenticating the client based on the received signed common nonce.

32. (Currently Amended) A computer program product for authenticating a client, comprising:

a computer readable media having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code which receives at a server of a plurality of servers, a common nonce which is associated with each of the plurality of servers from an entity other than the client or the plurality of servers, the common nonce being signed by the client; and

computer readable program code which authenticates the client based on the received signed common nonce.